

(19) 世界知的所有権機関
国際事務局



(43) 国際公開日
2003 年10 月9 日 (09.10.2003)

PCT

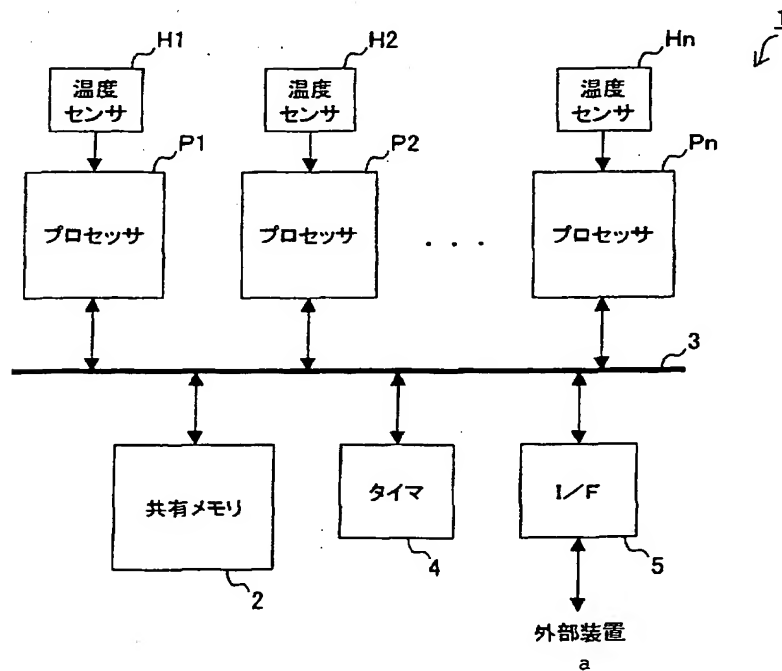
(10) 国際公開番号
WO 03/083693 A1

- (51) 国際特許分類⁷: G06F 15/177, 9/46, 1/00 (72) 発明者; および
(21) 国際出願番号: PCT/JP02/03324 (75) 発明者/出願人 (米国についてののみ): 平井 聡 (HIRAI, Akira) [JP/JP]; 〒211-8588 神奈川県 川崎市中
(22) 国際出願日: 2002 年4 月3 日 (03.04.2002) 区 上小田中4丁目1番1号 富士通株式会社内 Kanagawa
(25) 国際出願の言語: 日本語 (JP). 久門 耕一 (KUMON, Kouichi) [JP/JP]; 〒211-8588
(26) 国際公開の言語: 日本語 神奈川県 川崎市中区 上小田中4丁目1番1号 富士
(71) 出願人 (米国を除く全ての指定国について): 富士通 通株式会社内 Kanagawa (JP).
株式会社 (FUJITSU LIMITED) [JP/JP]; 〒211-8588 神 奈川 県 川 崎 市 中 原 区 上 小 田 中 4 丁 目 1 番 1 号 Kanagawa (JP).
(74) 代理人: 林 恒徳, 外 (HAYASHI, Tsunenori et al.); 〒 222-0033 神奈川県 横浜市港北区 新横浜3-9-5 第三東
昇ビル 林・土井国際特許事務所 Kanagawa (JP).
(81) 指定国 (国内): JP, US.

[続葉有]

(54) Title: TASK SCHEDULER IN DISTRIBUTED PROCESSING SYSTEM

(54) 発明の名称: 分散処理システムにおけるタスクスケジューリング装置



H1...TEMPERATURE SENSOR
P1...PROCESSOR
H2...TEMPERATURE SENSOR
P2...PROCESSOR

Hn...TEMPERATURE SENSOR
Pn...PROCESSOR
2...SHARED MEMORY
4...TIMER
a...EXTERNAL UNIT

(57) Abstract: A task scheduler for distributed processing system having a plurality of processors for processing a plurality of tasks while distributing. As a first scheduling method, the scheduler assigns a task to a processor of lowest temperature. As a second scheduling method, the scheduler selects a task based on the temperature of each processor and the characteristic value of the task related to the extent of temperature rise or increase in power consumption of each processor incident to execution of each task, and assigns a selected task to the processor. For example, a low temperature processor is assigned with a task of high temperature rise rate (e.g. a task having a large number of instructions being processed per unit time), as a second scheduling method. Temperature of respective processors can be made uniform by such a scheduling method.

(57) 要約: 本発明は、複数のタスクを分散して処理する複数の処理装置を有する分散処理システムのタスクスケジューリング装置を提供する。このタスクスケジューリング装置は、第1のタスクスケジューリング方法として、温度の最も低い処理装置にタスクを割り

当てる。また、タスクスケジューリング装置は、第2のタスクスケ

[続葉有]



添付公開書類:
— 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

ジューリング方法として、各処理装置の温度と、各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値とに基づいて、タスクを選択し、選択したタスクを処理装置に割り当てる。たとえば、第2のタスクスケジューリング方法として、温度の低い処理装置には、温度上昇の度合いの大きなタスク（たとえば単位時間あたりに処理される命令数の多いタスク）が割り当てられる。このようなスケジューリング方法により、各処理装置の温度を均一にすることができる。

明細書

分散処理システムにおけるタスクスケジューリング装置

5 技術分野

本発明は、タスクスケジューリング装置およびタスクスケジューリング方法に関し、特に、複数のタスクを分散して処理する複数の処理装置を有する分散処理システムのタスクスケジューリング装置およびタスクスケジューリング方法に関する。また、本発明は、タスクスケジューリングをコンピュータに実行させるプログラム
10 に関する。

背景技術

近年、CPU、MPU等のプロセッサの著しい性能向上に伴い、プロセッサの消費電力が増大し、これによりプロセッサの発熱量が増加している。その結果、プロ
15 セッサの温度上昇が問題となっている。

このため、プロセッサの温度上昇を防ぐために、プロセッサにファンを装着したり、プロセッサを格納した筐体内の風流を最適化したりする等の熱対策が講じられている。しかし、プロセッサの性能向上に伴う最大TDP（Thermal Design Power：熱設計電力）の増加により、ファンの大型化、消費電力の増大、および筐
20 体容積の増加を招いており、その結果、装置全体のコストが増加し、また装置が大型化するという問題が発生している。

また、プロセッサの電圧または周波数を制御する機構を設け、この機構により、プロセッサの電圧や周波数を、必要に応じて低減する熱対策も講じられている。しかし、この対策によると、プロセッサの処理能力が低下し、好ましくない。

25 一方で、プロセッサやプロセッサを有するコンピュータ等の処理装置を複数設けて、タスクを複数の処理装置に分散して処理させることにより、処理の負荷分散または機能分散を行い、処理の高速化を図る分散処理システムないしは並列処理システムが実用化されてきている。

このようなシステムでは、複数の処理装置が稼動するので、熱対策はより重要となるが、複数の処理装置が存在するために、特有の問題も生じている。すなわち、
30

- ある特定のプロセッサまたはプロセッサ群の温度上昇が他のプロセッサに対して大きくなる場合があり、しかも、そのようなプロセッサ（群）は、その時々処理において変化し一定でないために、結果的に、全プロセッサに対して上記のようなファンを取り付ける必要があり、大幅なコストの増加を招く。また、電圧や周波数を
- 5 低減させる方法では、処理の高速化を図るために、複数の処理装置を設けたマルチプロセッサ構成や分散コンピューティング環境の意味がなくなる。

発明の開示

- 本発明は、このような状況に鑑みなされたものであり、その目的は、複数の処理
- 10 装置を有する分散処理システムにおいて、各処理装置の温度をほぼ均等にするためのタスクスケジューリング装置およびタスクスケジューリング方法を提供することにある。

- 前記目的を達成するために、本発明の第1の側面によるタスクスケジューリング装置は、複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムの前記各処理装置への
- 15 タスクスケジューリングを実行するタスクスケジューリング装置であって、前記計測装置により計測された各処理装置の温度または消費電力を比較する比較部と、前記比較部の比較の結果、前記計測装置により計測された温度または消費電力が最も低い処理装置にタスクを割り当てるタスク割り当て部と、を備えている。

- 20 本発明の第1の側面によるタスクスケジューリング方法は、複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムにおける前記複数の処理の少なくとも1つにより、または、前記複数の処理装置とは別個に設けられた制御装置により実行されるタスクスケジューリング方法であって、前記計測装置により計測された各処理装置の温
- 25 度または消費電力を比較し、前記比較の結果、前記計測装置により計測された温度または消費電力が最も低い処理装置にタスクを割り当てるものである。

- 本発明の第1の側面によるプログラムは、前記第1の側面によるタスクスケジューリング方法を、複数のタスクを分散して処理する処理装置の少なくとも1つまたは前記複数の処理装置とは別個に設けられた制御装置に設けられたコンピュータに
- 30 実行させるものである。

また、本発明の第1の側面による分散処理システムは、複数のタスクを分散して処理する複数の処理装置を有する分散処理システムであって、前記複数の処理装置のそれぞれの温度または消費電力を計測する計測装置と、前記複数の処理装置とは別装置として設けられ、または、前記複数の処理装置の少なくとも1つに設けられ、
5 前記計測装置により計測された各処理装置の温度または消費電力を比較し、前記比較の結果、前記計測装置により計測された温度または消費電力が最も低い処理装置にタスクを割り当てるタスクスケジューリング装置と、を備えている。

ここで、前記タスクスケジューリング装置は、前記複数の処理装置の少なくとも1つが備えていてもよいし、前記複数の処理装置とは別個の装置として設けられて
10 もよい。

本発明の第1の側面によると、温度または消費電力の最も低い処理装置にタスクが割り当てられるので、温度または消費電力の最も低い処理装置はタスクの処理に伴い発熱し、温度が上昇する一方、温度または消費電力の高い他の処理装置は、タスクが割り当てられないことにより、発熱量が減少する。その結果、各処理装置の
15 温度を均一にさせて行くことができる。

本発明の第2の側面によるタスクスケジューリング装置は、複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムの前記各処理装置へのタスクスケジューリングを実行するタスクスケジューリング装置であって、各タスクの実行に伴う各処理装置の
20 温度上昇または消費電力増加の度合いに関連するタスクの特性値をタスクごとに記憶する記憶部と、タスクの割り当て対象となる処理装置についての、前記計測装置により計測された温度または消費電力と、前記記憶部に記憶された前記特性値とに基づいて、前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタスクから選択し、該選択したタスクを前記タスクを割り当てる対象となる処理装置に割り当てるタスク割り当て部と、を備えている。
25

本発明の第2の側面によるタスクスケジューリング方法は、複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムにおける前記複数の処理の少なくとも1つにより、または、前記複数の処理装置とは別個に設けられた制御装置により実行されるタスクスケジューリング方法であって、タスクの割り当て対象となる処理装置について
30

の、前記計測装置により計測された温度または消費電力と、内部のメモリまたは外部の共有メモリに記憶され、各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値とに基づいて、前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタスクから選択し、該選択したタスクを前記タスクを割り当てる対象となる処理装置に割り当てるものである。

本発明の第2の側面によるプログラムは、前記第2の側面によるタスクスケジューリング方法を、複数のタスクを分散して処理する処理装置の少なくとも1つまたは前記複数の処理装置とは別個に設けられた制御装置に設けられたコンピュータに実行させるものである。

10 本発明の第2の側面による分散処理システムは、複数のタスクを分散して処理する複数の処理装置を有する分散処理システムであって、前記複数の処理装置のそれぞれの温度または消費電力を計測する計測装置と、各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値をタスクごとに記憶する記憶装置と、タスクの割り当て対象となる処理装置についての、前記計測装置により計測された温度または消費電力と、前記記憶装置に記憶された前記特性値とに基づいて、前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタスクから選択し、該選択したタスクを前記タスクを割り当てる対象となる処理装置に割り当てるタスク割り当て部と、を備えている。

ここで、前記タスクスケジューリング装置は、前記複数の処理装置の少なくとも1つが備えていてもよいし、前記複数の処理装置とは別個の装置として設けられてもよい。

本発明の第2の側面によると、各処理装置の温度または消費電力と、各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値とに基づいて、たとえば温度または消費電力の大きな処理装置には、温度上昇または消費電力増加の度合いが小さいことを示す特性値を有するタスクを割り当て、温度または消費電力の小さな処理装置には、この逆を行うことにより、各処理装置の温度を均一にさせて行くことができる。

図面の簡単な説明

30 図1は、本発明の第1の実施の形態による分散処理システムの構成例を示すプロ

ック図である。

図2は、共有メモリに記憶されたデータを示す。

図3は、各プロセッサにより実行される第2のタスクスケジューリング方法の処理の流れを示すフローチャートである。

- 5 図4は、各プロセッサにより実行される第3のタスクスケジューリング方法の処理の流れを示すフローチャートである。

図5は、本発明の第2の実施の形態による分散処理システムの構成例を示すブロック図である。

10 発明を実施するための最良の形態

<第1の実施の形態>

- 図1は、本発明の第1の実施の形態による分散処理システム1の構成例を示すブロック図である。この分散処理システム1は、たとえば1つの筐体に収納されるマルチプロセッサシステムであり、n個（nは2以上の整数、以下同じ。）のプロセッサP1～Pn、n個の温度センサH1～Hn、共有メモリ2、バス3、タイマ4、および通信インタフェース装置（I/F）5を備えている。

- プロセッサP1～Pn、共有メモリ2、タイマ4、およびI/F5は、バス3に接続されている。プロセッサP1～Pnは、バス3を介して、共有メモリ2に記憶されたプログラムまたはデータを読み出し、また、処理により生成されたデータまたはプログラムを共有メモリ2に書き込むことができるようになっている。

- プロセッサP1～Pnは、たとえばCPU、MPU等、または、CPU、MPU等とその周辺のハードウェア回路とを備えた装置（たとえばプロセッサボード）により構成される。このプロセッサP1～Pnは、内部にメモリ（キャッシュメモリを含む。）を有し、共有メモリ2に記憶されているOS（OSプログラム）およびアプリケーションプログラム（実行待ちタスクキューにあるタスクに対応するタスクプログラム）を実行する。

- また、プロセッサP1～Pnは、性能モニタリング機能を有する。この性能モニタリング機能を使用することにより、プロセッサP1～Pnは、実行した命令数、タスクの処理に要した時間またはクロック数、メモリへのアクセス回数、単位時間当たり
- 30 当りに実行した命令数、単位時間当たりのメモリへのアクセス回数、これらの組

み合わせ（たとえば単位時間当りに実行される命令数とメモリアクセス回数の合計値）等の性能に関する数値を計測することができる。いずれの数値を計測するかを、各プロセッサにあらかじめ指定しておくことができる。

5 タイマ4には、プロセッサP 1～P nのいずれかにより時間が設定され、タイマ4は、設定された時間が経過すると、タイマ割り込み信号をバス3に出力する。出力された割り込み信号は、プロセッサP 1～P nのいずれかにより受信され処理される。このタイマ4は、たとえば、所定の時間の間、タスクをスリープ状態に置き、所定の時間の経過後、このスリープ状態のタスクを目覚めさせ（ウェイクアップさせ）、プロセッサにより実行させる場合に利用される。

10 I/F 5は、この分散処理システム1の外部の装置（コンピュータ等）に接続され、この外部装置との間で通信インタフェースの処理（プロトコルの処理等）を実行する。このI/F 5は、外部装置からデータを受信すると、割り込み信号をバス3に出力する。出力された割り込み信号は、いずれかのプロセッサにより受信され処理される。

15 温度センサH 1～H nは、それぞれプロセッサP 1～P nの温度を計測するセンサである。プロセッサP i（iは1～nのいずれかの整数、以下同じ。）は、対応する温度センサH iの温度をあらかじめ定められた一定時間間隔で読み出し、読み出した温度を共有メモリの所定の領域（後述）に書き込む。

20 温度センサH 1～H nは、図1に示すように、プロセッサP 1～P nとは別に設けられてもよいし、プロセッサP 1～P nのハードウェア回路に内蔵されていてもよい。温度センサH 1～H nがプロセッサP 1～P nと別に設けられる場合には、プロセッサP 1～P nの表面に接触して取り付けられるか、あるいは、プロセッサP 1～P nの近傍に離間（数mmの間隔を離間）して設置されることとなる。また、プロセッサP 1～P nが温度センサH 1～H nの温度をそれぞれ読み出して、共有
25 メモリ2に温度を書き込むのではなく、温度センサH 1～H nがバス3に直接接続され、一定時間間隔で計測した温度を、バス3を介して共有メモリ2に書き込んでよい。

共有メモリ2は、たとえばRAMであり、OS（OSプログラム）、アプリケーションプログラム等を記憶する。図2は、共有メモリ2に記憶されたデータ（プログラムを含む。）を示している。共有メモリ2に記憶されたデータには、OS、アプリ
30

ケーションプログラム、プロセッサ温度データ、発熱イベント頻度データ、タスクキュー等が含まれる。

OSは、プロセッサP1～Pnにより実行される共通のOSであり、プロセッサP1～Pnは、このOSを読み出して実行する。このOSには、スケジューラ（スケジューリングプログラム）が含まれており、プロセッサP1～Pnは、このスケジューリングプログラムに従ってスケジューリングを実行する。後に詳述するように、このスケジューリングにおいて、本発明の第1の実施の形態によるタスクのスケジューリング処理（すなわちタスクの選択および割り当て処理）が実行される。

アプリケーションプログラムは、プロセッサP1～Pnによる実行単位であるタスク（タスクプログラム）に分割され、図2では、m個（mは2以上の整数、以下同じ。）のタスクプログラムK1～Kmに分割されている。これらのタスクプログラムK1～Kmが、プロセッサP1～Pnによって実行されることにより、機能分散および負荷分散の双方またはいずれか一方が図られ、アプリケーションプログラムの処理が高速化される。

タスクキューには、実行待ちのタスク（たとえばタスクプログラムを示す識別子等）がキューイングされている。プロセッサP1～Pnは、タスクキューに存在するタスクの中から1つを選択して自己または他のプロセッサに割り当てる。タスクを割り当てられたプロセッサは、割り当てられたタスクに対応するタスクプログラムを実行する。また、プロセッサP1～Pnは、タスクプログラムの実行によって新たにタスクが生成されると、このタスクをタスクキューにキューイングする。

プロセッサ温度データおよび発熱イベント頻度データは、プロセッサP1～Pnが、タスクスケジューリングを実行する際に、タスクの選択基準として、または、タスクを割り当てる対象となるプロセッサの選択基準として使用される。

プロセッサ温度データは、全プロセッサの平均温度TaおよびプロセッサP1～Pnのそれぞれの温度T1～Tnのデータ項目を有する。

温度T1～Tnは、前述した温度センサH1～Hnによりそれぞれ計測され、プロセッサP1～Pnにより一定時間間隔で書き込まれた温度である。したがって、温度T1～Tnは、一定時間間隔で更新される。

全プロセッサの平均温度Taは、温度T1～Tnの平均値である。すなわち、 $Ta = (T1 + T2 + \dots + Tn) \div n$ により求められる。この平均温度Taは、たと

えばプロセッサ P_i が温度 T_i を書き込んだ時に、書き込み後、プロセッサ P_i により温度 $T_1 \sim T_n$ に基づいて計算され、更新される。したがって、この平均温度 T_a は、各プロセッサ $P_1 \sim P_n$ が自己の温度を書き込む際に、更新されることとなる。

5 発熱イベント頻度データは、各プロセッサの温度上昇または消費電力増加の度合いに関連するタスクの特性値の一例である。この発熱イベント頻度データは、現在までの発熱イベント頻度の平均値（平均発熱イベント頻度） E_a およびタスクプログラム $K_1 \sim K_m$ のそれぞれについての発熱イベント頻度 $E_1 \sim E_m$ のデータ項目を有する。

10 ここで、「発熱イベント」とは、プロセッサに熱を発生させる原因となるイベントであり、たとえばプロセッサにより実行される命令、プロセッサの内部メモリまたは共有メモリ 2 へのアクセス等を発熱イベントとして挙げることができる。したがって、「発熱イベント頻度」としては、たとえば、タスクまたは OS の処理における単位時間あたりに実行される命令数、単位時間当りのメモリアクセス回数、これら
15 の組み合わせ（たとえば単位時間あたりに実行される命令数とメモリアクセス回数の合計値）等がある。

なお、発熱イベント頻度の代わりに、各タスクに含まれる命令数やメモリアクセス回数等の「発熱イベント数」を使用することもできる。また、タスクの処理に要する時間をタスクの選択基準として使用することもできる。

20 本実施の形態では、一例として「発熱イベント頻度」を使用し、また、発熱イベント頻度として、各タスクの実行時に、単位時間あたりに実行される命令数を使用することとする。すなわち、タスクプログラム $K_1 \sim K_m$ のそれぞれについて、実行された命令数を $I_1 \sim I_m$ とし、これらの命令数をそれぞれ実行するのに要する時間（またはクロック数）を $t_1 \sim t_m$ とすると、発熱イベント頻度 $E_1 \sim E_m$ は、
25 $E_1 = I_1 \div t_1, \dots, E_i = I_i \div t_i, \dots, E_m = I_m \div t_m$ となる。なお、時間としてクロック数を使用した場合に、発熱イベント頻度の単位は IPC (Instruction per Clock) となる。

たとえば、タスクプログラム K_j (j は $1 \sim m$ のいずれかの整数、以下同じ。) がプロセッサ P_i により実行されると、プロセッサ P_i は、性能モニタリング機能を使用して、タスクプログラム K_j の実行した命令数および実行に要する時間を計測
30

し、これら命令数および時間から発熱イベント頻度 E_j を計算する。そして、プロセッサ P_i は、発熱イベント頻度 E_j を共有メモリ 2 に書き込む。

これらの命令数や時間は、プロセッサ P_i がタスクプログラム K_j の実行開始時に性能モニタリング機能から読み出した値と、実行終了時に性能モニタリング機能から読み出した値との差分により求めることもできるし、プロセッサ P_i がタスクプログラム K_j の実行開始時に性能モニタリング機能の値を 0 にリセットし、実行終了時に性能モニタリング機能から読み出した値により求めることもできる。

また、同じタスクプログラム K_j であっても、ある時点で実行された場合の発熱イベント頻度の値と、他の時点で実行された場合の発熱イベント頻度の値とが異なる場合がある。たとえば、タスクプログラム K_j が条件分岐や繰り返しループを有する場合に、ある時点で実行されたときの、選択された分岐や繰り返しループ回数と、他の時点で実行されたときの、選択された分岐や繰り返しループ回数とが異なる場合に、このような発熱イベント頻度の値が異なる事態が生じる。

したがって、発熱イベント頻度 E_j としては、(a) タスクプログラム K_j が最も近時に実行された時点の値とすることもできるし、(b) タスクプログラム K_j がこれまでに実行されたすべての場合の平均発熱イベント頻度値とすることもできる。

前者 (a) の場合に、プロセッサ P_i は、タスクプログラム K_j の実行後、性能モニタリング機能により求められた発熱イベント頻度 E_j を共有メモリ 2 の所定のアドレスに書き込む (上書きする) だけでよい。

20 後者 (b) の場合に、図 2 における図示は省略するが、共有メモリ 2 には、タスクプログラム K_j がこれまでに実行されたすべての場合の命令数の合計値 ($I_{j_{a11}}$ とする。) と、実行に要した時間の合計値 ($t_{j_{a11}}$ とする。) とが記憶される。たとえば、タスクプログラム K_j がこれまでに x 回実行されている場合には、 $I_{j_{a11}} = I_{j_1} + I_{j_2} + \dots + I_{j_x}$ 、 $t_{j_{a11}} = t_{j_1} + t_{j_2} + \dots + t_{j_x}$ となる (I_{j_k} (25 k は $1 \sim x$ のいずれかの整数) はタスクプログラム K_j が第 k 回目に実行された場合の命令数、 t_{j_k} はタスクプログラム K_j が第 k 回目に実行された場合の実行時間)。そして、命令数の合計値を時間の合計値で割った値が発熱イベント頻度 E_j とされる。すなわち、 $E_j = I_{j_{a11}} \div t_{j_{a11}}$ となる。

たとえば、プロセッサ P_i が、第 $(x+1)$ 回目のタスクプログラム K_j を実行した場合には、実行後、性能モニタリング機能により求められた命令数 $I_{j_{x+1}}$ およ

び実行時間 $t_{j_{x+1}}$ を、共有メモリ 2 に記憶された $I_{j_{a11}}$ および $t_{j_{a11}}$ にそれぞれ加算し、加算後の値に基づいて $E_j = I_{j_{a11}} \div t_{j_{a11}}$ を計算し、計算した E_j を新たな発熱イベント E_j として共有メモリ 2 に書き込む（上書きする）こととなる。

- 5 なお、タスクプログラムが実行されないと、発熱イベント頻度は求められないので、タスクプログラムが 1 度も実行されていない時点における発熱イベント頻度 $E_1 \sim E_m$ の値（すなわち初期値）は、あらかじめ定められた値とされる。この初期値は、たとえばタスクプログラム $K_1 \sim K_m$ を実験やシミュレーションにより実行して求めた値とすることができる。

- 10 「現在までの平均発熱イベント頻度 E_a 」は、全プロセッサ $P_i \sim P_n$ により、これまでに実行されたすべてのタスクの平均発熱イベント頻度値である。

すなわち、平均値 E_a は、タスクプログラム $K_1 \sim K_m$ のそれぞれについてのこれまでの実行命令数の合計値の総和（ $I_{a11} = I_{1_{a11}} + I_{2_{a11}} + \dots + I_{m_{a11}}$ ）を、これまでの実行命令数の合計値の総和（ $t_{a11} = t_{1_{a11}} + t_{2_{a11}} + \dots + t_{m_{a11}}$ ）で割った値（ $E_a = I_{a11} \div t_{a11}$ ）とされる。

- 15 E_a ）で割った値（ $E_a = I_{a11} \div t_{a11}$ ）とされる。

なお、各タスクプログラムの毎回の実行時間を一定であるとみなすと、平均値 E_a は以下の式で表すこともできる。

$$E_a = \{(E_{1_1} + E_{1_2} \dots + E_{1_{n_1}}) + (E_{2_1} + E_{2_2} \dots + E_{2_{n_2}}) + \dots + (E_{j_1} + E_{j_2} \dots + E_{j_{n_j}}) + \dots + (E_{m_1} + E_{m_2} \dots + E_{m_{n_m}})\} \div (n_1 + n_2 + \dots + n_j + \dots + n_m)$$

- 20 $+ n_j + \dots + n_m$)

ここで、タスク K_j は n_j 回実行され、第 1 回から第 n_j 回までのそれぞれの発熱イベント頻度を $E_{j_1} \sim E_{j_{n_j}}$ としている。

プロセッサ P_i は、タスクプログラム K_j の実行後、タスクプログラム K_j の発熱イベント頻度 E_j を更新するとともに、平均発熱イベント頻度 E_a の値も計算し、

- 25 計算した値により、共有メモリ 2 の値を更新する。

このようなマルチプロセッサシステム 1 において、プロセッサ $P_1 \sim P_n$ は、これまで実行していたタスク（タスクプログラム）の処理が終了したり、タスクの切り替えが発生したり、あるいは、タイマ 4 または I/F 5 から割り込みが発生したりすると、タスクキューにキューイングされた実行待ちのタスクから 1 つを選択し、

- 30 選択したタスクを自己または他のプロセッサに割り当てるタスクスケジューリング

を実行する。このタスクスケジューリングの方法には、以下の方法がある。

(1) 第1のタスクスケジューリング方法

第1のタスクスケジューリング方法は、タスクを実行していないアイドル状態のプロセッサが複数存在する場合に、アイドル状態のプロセッサの温度にのみ基づいて、温度の最も低いプロセッサにタスクを割り当てるものである。

たとえば、タイマ4が、設定された時間の経過により割り込み信号を発生し、この割り込み信号がプロセッサP_iにより受信されると、プロセッサP_iは、これまで実行していたタスクを一時的に中断し、スケジューラを実行する。あるいは、プロセッサP_iがアイドル状態にある場合には、割り込み信号の受信により直ちにスケジューラを実行する。

プロセッサP_iは、割り込み信号受信時におけるアイドル状態のプロセッサが複数あるかどうかを判断する。プロセッサP_i自身もアイドル状態にあるならば、自己も対象となる。プロセッサがアイドル状態にあるかどうかは、プロセッサP_iが各プロセッサに問い合わせることによって確認することもできるし、各プロセッサが共有メモリ2の所定の領域に自己の状態（アイドル状態またはタスク処理状態）を書き込む場合には、この領域を読み出すことによって判断することもできる。

続いて、アイドル状態のプロセッサが複数存在する場合に、プロセッサP_iは、アイドル状態にあるプロセッサの温度を共有メモリ2から読み出し、温度の最も低いプロセッサを選択する。温度の最も低いプロセッサが複数存在する場合には、たとえば擬似乱数等が発生させ、発生された数値に基づいて1つのプロセッサを選択することができる。

続いて、プロセッサP_iは、選択したプロセッサに、ウェイクアップ状態に移行するタスクを割り当てる。

I/F5から割り込み信号がプロセッサP_iに入力された場合にも、プロセッサP_iは、同様にして、アイドル状態にあるプロセッサのうち、温度の最も低いプロセッサを選択し、選択したプロセッサにタスク（たとえばI/F5からのデータ受信処理等）を割り当てることができる。

このように、アイドル状態にあるプロセッサのうち、温度の最も低いプロセッサにタスクが割り当てられ、実行されるので、各プロセッサの発熱量が均等化されて行き、その結果、各プロセッサの温度を均一にすることができる。

なお、アイドル状態のプロセッサが1つの場合には、このプロセッサにタスクを割り当てることもできるし、他のプロセッサで温度の最も低いものに、割り当てることもできる。また、アイドル状態のプロセッサが存在しない場合にも、温度の最も低いプロセッサにタスクを割り当てることもできる。アイドル状態でないプロセッサにタスクが割り当てられた場合には、割り当てられるタスクの優先順位が実行中のタスクの優先順位より高い場合には、実行中のタスクが中断され、新たに割り当てられたタスクを実行することもできる。

(2) 第2のタスクスケジューリング方法

第2のタスクスケジューリング方法は、プロセッサの温度および発熱イベント頻度の双方に基づいてタスクを選択し、割り当てるものである。図3は、各プロセッサにより実行される第2のタスクスケジューリング方法の処理の流れを示すフローチャートである。この処理は、前述したように、OSのスケジューラの一部である。

プロセッサ P_i において、これまで実行していたタスクの処理が終了し、または、タスクの切り替えが実行されると、プロセッサ P_i は、共有メモリ2にアクセスして、共有メモリ2のタスクキューに複数のタスクが存在するかどうかを判断する(S1)。

タスクキューに複数のタスクが存在する場合には(S1でYES)、プロセッサ P_i は、自己の温度 T_i と共有メモリ2に記憶された平均温度 T_a とを比較する(S2)。ここで、自己の温度 T_i は、共有メモリ2に記憶されたものを使用することもできるし、プロセッサ P_i が、この比較を行う時に、温度センサ H_i から読み出したものを使用することもできる。

$T_i > T_a$ であるならば(S2でYES)、プロセッサ P_i は、タスクキューに存在する各タスクの発熱イベント頻度 E および平均発熱イベント頻度 E_a を共有メモリ2から読み出し、各タスクの発熱イベントと平均発熱イベント頻度 E_a とをそれぞれ比較する(S3)。そして、プロセッサ P_i は、平均発熱イベント頻度 E_a 以下の発熱イベント頻度 E (すなわち $E \leq E_a$)を有するタスクがタスクキューに存在するかどうかを判断する(S3)。

$E \leq E_a$ となるタスクがタスクキューに存在する場合には(S2でYES)、プロセッサ P_i は、 $E \leq E_a$ となるタスクからタスクを1つ選択し(S4)、選択したタスクを実行する。 $E \leq E_a$ となるタスクが1つの場合には、そのタスクが選択され

る。

ここで、 $E \leq E_a$ となるタスクが複数存在する場合には、その中で最小の発熱イベント頻度を有するタスクを選択することもできるし、その中で最大の発熱イベント頻度を有するタスクを選択することもできるし、中程度の発熱イベント頻度を有するタスクを選択することもできる。あるいは、擬似乱数等の数値を発生させ、この数値に基づいてタスクを選択することもできる。また、通常のスケジューリングと同様に、タスクの優先順位に基づいて最も優先順位の高いタスクを選択することもできる。さらに、同じ優先順位を有するタスクが複数存在する場合には、その中から、タスクキューの先頭位置により近いタスクまたはタスクキューに時間的に先にキューイングされたタスクを選択することもできる。

一方、 $E \leq E_a$ となるタスクがタスクキューに存在しない場合には(S 3でNO)、プロセッサP iは、タスクキューに存在するタスクのうち、最小の発熱イベント頻度Eを有するタスクを選択し(S 5)、選択したタスクを実行する。

ステップS 2において、 $T_i \leq T_a$ である場合に(S 2でNO)、プロセッサP iは、タスクキューに存在するタスクの中で、平均発熱イベント頻度 E_a より大きな発熱イベント頻度E($E > E_a$)を有するタスクが存在するかどうかを判断する(S 6)。

$E > E_a$ となるタスクがタスクキューに存在する場合には(S 6でYES)、プロセッサP iは、 $E > E_a$ となるタスクの中からタスクを1つ選択し(S 7)、選択したタスクを実行する。 $E > E_a$ となるタスクが1つの場合には、そのタスクが選択される。

$E > E_a$ となるタスクが複数存在する場合には、前述したのと同様に、その中から最大の発熱イベント頻度、最小の発熱イベント頻度、または中程度の発熱イベント頻度を有するタスクを選択したり、擬似乱数等の数値に基づいて選択したり、あるいは、通常のスケジューリングと同様の選択処理によりタスクを選択することができる。

$E > E_a$ となるタスクがタスクキューに存在しない場合には(S 6でNO)、プロセッサP iは、タスクキューに存在するタスクの中で最大の発熱イベント頻度Eを有するタスクを選択し(S 8)、選択したタスクを実行する。

ステップS 1において、タスクキューに複数のタスクが存在しない場合には、プ

ロセッサ P_i は、さらにタスクキューに存在するタスクが1つかどうかを判断する (S9)。タスクキューに存在するタスクが1つならば (S9でYES)、プロセッサ P_i はそのタスクを選択して (S10)、実行し、タスクキューにタスクが存在しない場合には (S9でNO)、プロセッサ P_i は、アイドルタスクを実行する。

- 5 なお、選択されたタスクは、タスクキューから消去される。また、プロセッサ P_i は、タスクキューにタスクが存在しない場合に、アイドルタスクを実行するのではなく、自己を停止状態にすることもできる。この場合には、タスクキューにタスクが発生した時点で、他の稼動状態にあるプロセッサによって、プロセッサ P_i は停止状態から稼動状態にされることとなる。

- 10 このように第2のタスクスケジューリング方法によると、プロセッサ P_i の温度 T_i が平均温度 T_a と比較され、温度 T_i が平均温度 T_a 以下である場合には、タスクキューに存在するタスクのうち、なるべく大きな発熱イベント頻度を有するタスクが選択される。したがって、選択されたタスクをプロセッサ P_i が実行することにより発生する発熱量は、一般に、平均的な発熱量よりも大きくなる。一方、温度 T_i が平均温度 T_a より大きな場合には、なるべく小さな発熱イベント頻度を有するタスクが選択される。したがって、選択されたタスクをプロセッサ P_i が実行することにより発生する発熱量は、一般に、平均的な発熱量よりも小さくなる。

- 15 これにより、各プロセッサの発熱量が均等化されて行き、その結果、各プロセッサの温度が均一化されるので、ある特定のプロセッサ (群) のみが高温になることを防止することができる。その結果、各プロセッサに大規模なファンを取り付けたり、熱設計のために大きな筐体を設けたりする必要が回避され、システムのコスト増大および大型化が防止できる。また、プロセッサの電圧や周波数を抑制することも回避でき、各プロセッサの処理能力を最大限活用することもできる。

- 20 なお、ステップS2における比較 $T_i > T_a$ は、 $T_i \geq T_a$ であってもよいし、ステップS3における比較を $E < E_a$ とし、ステップS6における比較を $E \geq E_a$ としてもよい。

(3) 第3のタスクスケジューリング方法

- 第3のタスクスケジューリング方法も、第2のタスクスケジューリング方法と同様に、プロセッサの温度および発熱イベント頻度に基づいてタスクの選択および割り当てを行うものである。図4は、各プロセッサにより実行される第3のタスクス
- 30

ケジューリング方法の処理の流れを示すフローチャートである。この処理は、前述したように、OSのスケジューラの一部である。

- プロセッサP_iは、タスクキューに複数のタスクが存在するかどうかを判断する(S 2 1)。タスクキューに複数のタスクが存在する場合には(S 2 1でYES)、
- 5 プロセッサP_iは、共有メモリ2に記憶されたプロセッサ温度データに基づいて、全プロセッサの温度T₁~T_nにおける自己の温度T_iの、温度の低いものからの順位(順位rとする。)を求める(S 2 2)。

- 続いて、プロセッサP_iは、タスクキューに存在するタスクを、発熱イベント頻度に基づいて発熱イベント頻度の値の大きなものから小さなものに向けてソートする(S 2 3)。
- 10

次に、プロセッサP_iは、ステップS 2 2で求めた自己の温度T_iの順位rに対応した発熱イベント頻度を有するタスクの中から1つのタスクを選択し(S 2 4)、選択したタスクを実行する。

- ここで、自己の温度T_iの順位rに対応した発熱イベント頻度は、たとえば、次のようにして決定される。まず、プロセッサP_iは、タスクキューに存在するタスクを、発熱イベント頻度の大きなものから小さなものに向けてn個(すなわちプロセッサP₁~P_nの個数)のグループG₁~G_nに分割する。そして、プロセッサP_iは、自己の温度の順位rに対応するグループG_rに属するタスクの中からタスクを1つ選択する。すなわち、自己の温度の順位が全プロセッサにおいて、低いものからr番目に位置する場合には、発熱イベント頻度の大きなものからr番目のグループG_rからタスクが選択される。
- 15
- 20

- これにより、相対的に温度の低いプロセッサには、相対的に発熱イベント頻度の高いタスクが割り当てられ、相対的に温度の高いプロセッサには、相対的に発熱イベント頻度の低いタスクが割り当てられる。その結果、各プロセッサの発熱量が平均化され、各プロセッサの温度が均一化されるので、ある特定のプロセッサ(群)のみが高温になることを防止することができる。その結果、プロセッサに大規模なファンを取り付けたり、熱設計のために大きな筐体を設けたりする必要が回避され、システムのコスト増大および大型化が防止できる。また、プロセッサの電圧や周波数を抑制することも回避でき、各プロセッサの処理能力を最大限活用することもできる。
- 25
- 30

なお、タスクキューに存在するタスクの個数（個数 p とする。）がプロセッサの個数 n 未満（すなわち $p < n$ ）である場合には、タスクキューに存在するタスクを n 個のグループに分割するのではなく、温度の順位を、温度の低いものから高いものに向けて p 個のグループ $G_1 \sim G_p$ に分割し、プロセッサ P_i の温度 T_i が属する

5 グループ G_r に対応するタスク T_r が選択される。すなわち、プロセッサ P_i の温度 T_i が、温度の低いものから r 番目のグループ G_r に属する場合には、発熱イベント頻度の高いものから r 番目のタスクが選択される。

これによっても、各プロセッサの発熱量が平均化され、各プロセッサの温度が均一化されるので、ある特定のプロセッサ（群）のみが高温になることを防止することができるとは言うまでもない。

10

一方、ステップ S_{21} で、タスクキューに複数のタスクが存在しない場合には、プロセッサ P_i は、ステップ S_{25} および S_{26} の処理を実行する。これらステップ S_{25} および S_{26} は、前述した図3のステップ S_9 および S_{10} とそれぞれ同じであるので、ここではその説明を省略する。

15 <第2の実施の形態>

図5は、本発明の第2の実施の形態による分散処理システム10の構成例を示すブロック図である。この分散処理システム10は、分散コンピューティングシステムであり、コントローラ11、 n 個のノード $N_1 \sim N_n$ 、および通信ネットワーク12を備えている。

20 ノード $N_1 \sim N_n$ およびコントローラ11は、通信ネットワーク12に接続され、通信ネットワーク12を介して相互に通信可能となっている。通信ネットワーク12は、たとえばLAN、インターネット等である。

ノード $N_1 \sim N_n$ のそれぞれは、たとえばコンピュータであり、内部に、CPU、MPU等により構成されるプロセッサ21、通信ネットワーク12との通信インタフェース処理を実行する通信インタフェース装置（I/F）22、およびプロセッサ21の温度を測定する温度センサ23を有する。

25

コントローラ11は、たとえばコンピュータであり、その内部の記憶装置（図示略）には、前述した図2に示す共有メモリ2のデータと同じデータが記憶されている。すなわち、内部の記憶装置には、スケジューラを含むOS、アプリケーションプログラム、プロセッサ温度データ、発熱イベント頻度データ、タスクキュー等が

30

記憶されている。

また、コントローラ 11 は、内部にタイマを有し、タイマの割り込み信号によって、所定のスリープ状態にあるタスクをウェイクアップ状態にし、このタスクをいずれかのノードに割り当てて実行させることもできる。

- 5 本実施の形態において、コントローラ 11 は、タスクスケジューリングを専用に行い、自らはタスクを実行しない。このため、コントローラ 11 は、内部の記憶装置に記憶されたスケジューラを実行して、ノード N1 ~ Nn のタスクスケジューリング処理を実行する。

- 10 タスクスケジューリング処理においては、ノード N1 ~ Nn がタスク割り当て要求をコントローラ 11 に送信し、コントローラ 11 がこの要求に応じて、要求を送信したノードに対してタスクを選択し割り当ててもよいし、コントローラ 11 がアイドル状態のノードに対してタスクを選択し割り当ててもよい。ノードがアイドル状態かどうかは、ノード N1 ~ Nn からコントローラ 11 に送信される状態通知によって検知することもできるし、コントローラ 11 が定期的にノード N1 ~ Nn の
15 状態をチェックすることによっても検知することができる。

- 20 プロセッサ温度データにおける温度 T1 ~ Tn は、本実施の形態では、ノード N1 ~ Nn のそれぞれのプロセッサ 21 の温度である。前述した第 1 の実施の形態と同様に、各ノードのプロセッサ 21 は、温度センサ 23 により計測された自己の温度を一定時間間隔で読み出し、読み出した温度を I/F 22 および通信ネットワーク 12 を介してコントローラ 11 に送信する。コントローラ 11 は、各ノードから送信された温度を内部の記憶装置に記憶する。

また、平均温度 Ta は、コントローラ 11 が、温度 T1 ~ Tn に基づいて計算する。コントローラ 11 は、温度 T1 ~ Tn の少なくとも 1 つがノードから送信され、更新（記憶）されるごとに、更新後の値に基づいて平均温度 Ta を求める。

- 25 タスクプログラム K1 ~ Km のそれぞれの発熱イベント頻度 E1 ~ Em は、ノード N1 ~ Nn のそれぞれのプロセッサ 21 の性能モニタリング機能により計測された実行命令数、処理時間（またはクロック数）等によって求められた発熱イベント頻度の値である。各ノードのプロセッサ 21 は、あるタスクがコントローラ 11 から割り当てられ、割り当てられたタスクを実行すると、実行後、性能モニタリング
30 機能により計測された実行命令数、処理時間等をコントローラ 11 に送信する。コ

ントローラ 11 は、各ノードから送信されたこれらの値に基づいて、第 1 の実施の形態と同様にして発熱イベント頻度を計算し、第 1 の実施の形態における方法 (a) または (b) により、内部の記憶装置に記憶 (更新) する。

- また、平均発熱イベント頻度 E_a は、前述した第 1 の実施の形態と同様にして、
- 5 コントローラ 11 により計算され、記憶される。

このような分散処理システム 10 において、コントローラ 11 は、前述した第 1 の実施の形態における第 1、第 2、または第 3 のタスクスケジューリング方法を実行することにより、タスクの選択および割り当てを行う。具体的には、次のようにタスクスケジューリング処理が実行される。

10 (1) 第 1 のタスクスケジューリング方法

- コントローラ 11 は、たとえば、内部のタイマの割り込み信号により、スリープ状態のタスクをウェイクアップ状態にしてノードに割り当てる場合に、アイドル状態のノードが存在するかどうかを確認する。アイドル状態のノードが複数存在する場合に、コントローラ 11 は、それらノードの中から、温度の最も低いプロセッサ
- 15 を有するノードを選択し、選択したノードに、ウェイクアップ状態に移行するタスクを割り当て、実行させる。

このように、アイドル状態にあるノードのうち、温度の最も低いノードにタスクが割り当てられ、実行されるので、各ノードのプロセッサの発熱量が均等化されて行き、その結果、各ノードの温度を均一にすることができる。

- 20 なお、アイドル状態のノードが 1 つの場合には、このノードにタスクを割り当てることもできるし、他のノードで温度の最も低いプロセッサを有するものに、割り当てることもできる。また、アイドル状態のノードが存在しない場合にも、温度の最も低いプロセッサを有するノードにタスクを割り当てることもできる。アイドル状態でないノードにタスクが割り当てられた場合には、割り当てられるタスクの優先順位が実行中のタスクの優先順位より高い場合には、実行中のタスクが中断され、
- 25 新たに割り当てられたタスクを実行することもできる。

(2) 第 2 のタスクスケジューリング方法

- コントローラ 11 は、ノード N_i からタスクの割り当て要求を受信すると、ノード N_i の温度 T_i 、平均温度 T_a 、平均発熱イベント頻度 E_a 、および各発熱イベント頻度 $E_1 \sim E_m$ に基づいて、図 3 に示すフローチャートの処理を実行し、ノード
- 30

ドN_iに割り当てるタスクを選択する。そして、コントローラ11は、選択したタスクをノードN_iに割り当てる。

5 コントローラ11がアイドル状態のノードN_iを検出し、このアイドル状態のノードN_iに対して、図3に示すフローチャートの処理によりタスクを選択し、選択したタスクを割り当ててもよい。

これにより、各プロセッサの発熱量が均等化されて行き、その結果、各プロセッサの温度が均一化されるので、ある特定ノードのプロセッサ（群）のみが高温になることを防止することができる。

（3）第3のタスクスケジューリング方法

10 コントローラ11は、ノードN_iからタスクの割り当て要求を受信すると、図4に示すフローチャートの処理を実行し、ノードN_iに割り当てるタスクを選択する。そして、コントローラ11は、選択したタスクをノードN_iに割り当てる。

15 コントローラ11がアイドル状態のノードN_iを検出し、このアイドル状態のノードN_iに対して、図3に示すフローチャートの処理によりタスクを選択し、選択したタスクを割り当ててもよい。

これにより、各プロセッサの発熱量が均等化されて行き、その結果、各プロセッサの温度が均一化されるので、ある特定ノードのプロセッサ（群）のみが高温になることを防止することができる。

<他の実施の形態>

20 第1および第2の実施の形態において、プロセッサの温度に代えてプロセッサの消費電力を、プロセッサ（ノード）の選択基準またはタスクの選択基準に用いることもできる。この場合には、各プロセッサに内蔵され、または、各プロセッサに取り付けられる消費電力計測回路が消費電力を計測し、共有メモリ2またはコントローラ11の内部メモリには、プロセッサ温度データの代わりに、各プロセッサの消費電力の累積値およびその平均値が記憶される。

25 また、第1および第2の実施の形態において、発熱イベント頻度を、実行されるすべての命令を対象にして求めるのではなく、発熱量（および消費電力）の多い浮動小数点演算命令のみを対象にして求めることもできる。

30 第2の実施の形態において、1つのノードが、図1に示すように、複数のプロセッサを有するマルチプロセッサシステムであってもよい。この場合には、コントロ

ーラ 1 1 は、各ノードのプロセッサごとにタスクを選択して割り当てることができる。

- 5 なお、第 1 の実施の形態に示すマルチプロセッサシステムにおいても、プロセッサ P 1 ~ P n とは別にコントローラを設け、このコントローラが、第 2 の実施の形態におけるコントローラ 1 1 の機能を実行して、プロセッサ P 1 ~ P n へのタスクスケジューリングを実行することもできる。

産業上の利用の可能性

- 10 本発明は、マルチプロセッサシステム、複数のコンピュータが通信ネットワークに接続された分散コンピューティングシステム等の分散処理システムに適用することができる。

- 15 本発明によると、分散処理システムの各処理装置（プロセッサ、コンピュータ等）の温度を均一にさせて行くことができる。その結果、各処理装置に大規模なファンを取り付けたり、熱設計のために大きな筐体を設けたりする必要が回避され、システムのコスト増大および大型化が防止できる。また、各処理装置の電圧や周波数を抑制することも回避でき、各処理装置の処理能力を最大限活用することもできる。

請求の範囲

1. 複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムの前記各処理装置へのタ
5 スクスケジューリングを実行するタスクスケジューリング装置であって、

前記計測装置により計測された各処理装置の温度または消費電力を比較する比較部と、

前記比較部の比較の結果、前記計測装置により計測された温度または消費電力が最も低い処理装置にタスクを割り当てるタスク割り当て部と、

10 を備えているタスクスケジューリング装置。

2. 請求の範囲第1項において、

前記タスクスケジューリング装置は、前記複数の処理装置の少なくとも1つに設けられ、自己または他の処理装置に前記タスクスケジューリングを実行する、

15 タスクスケジューリング装置。

3. 請求の範囲第1項または第2項において、

前記比較部は、前記複数の処理装置のうち、アイドル状態にある処理装置の温度または消費電力を比較する、

20 タスクスケジューリング装置。

4. 複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または消費電力を計測する計測装置とを有する分散処理システムの前記各処理装置へのタ
25 スクスケジューリングを実行するタスクスケジューリング装置であって、

各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値をタスクごとに記憶する記憶部と、

タスクの割り当て対象となる処理装置についての、前記計測装置により計測された温度または消費電力と、前記記憶部に記憶された前記特性値とに基づいて、

前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタ
30 スクから選択し、該選択したタスクを前記タスクを割り当てる対象となる処理装置

に割り当てるタスク割り当て部と、
を備えているタスクスケジューリング装置。

5. 請求の範囲第4項において、

- 5 前記特性値が、各タスクの単位時間あたりに処理される命令数を表すイベント
頻度であり、

前記タスク割り当て部は、前記タスクの割り当て対象となる処理装置の温度が
前記複数の処理装置の平均温度以上もしくは平均温度より高いか、または、前記
処理装置の消費電力が前記複数の処理装置の平均消費電力以上もしくは平均消費
10 電力よりも高い場合には、これまでに実行されたすべてのタスクのイベント頻度
の平均値以下または平均値より小さなイベント頻度を有するタスクを実行待ちタ
スクから選択して前記処理装置に割り当てる、
タスクスケジューリング装置。

- 15 6. 請求の範囲第5項において、

前記タスク割り当て部は、前記これまでに実行されたタスクのイベント頻度の
平均値以下または平均値より小さなイベント頻度を有するタスクが実行待ちのタ
スクの中に存在しない場合には、実行待ちのタスクの中から最小のイベント頻度
を有するタスクを前記処理装置に割り当てる、

- 20 タスクスケジューリング装置。

7. 請求の範囲第4項において、

前記特性値が、各タスクの単位時間あたりに処理される命令数を表すイベント
頻度であり、

- 25 前記タスク割り当て部は、前記タスクの割り当て対象となる処理装置の温度が
前記複数の処理装置の平均温度以下もしくは平均温度より小さいか、または、前
記処理装置の消費電力が前記複数の処理装置の平均消費電力以下もしくは平均消
費電力より小さい場合には、これまでに実行されたすべてのタスクのイベント頻
度の平均値以上または平均値より大きなイベント頻度を有するタスクを実行待ち
30 タスクから選択して前記処理装置に割り当てる、

タスクスケジューリング装置。

8. 請求の範囲第7項において、

- 5 前記タスク割り当て部は、前記これまでに実行されたタスクのイベント頻度の
平均値以下または平均値より小さなイベント頻度を有するタスクが実行待ちのタ
スクの中に存在しない場合には、実行待ちのタスクの中から最大のイベント頻度
を有するタスクを前記処理装置に割り当てる、

タスクスケジューリング装置。

10 9. 請求の範囲第4項において、

前記特性値が、各タスクの単位時間当たりに処理される命令数を表すイベント
頻度であり、

- 15 前記タスク割り当て部は、前記複数の処理装置における、前記タスクの割り当
て対象となる処理装置の温度の順位を求め、実行待ちタスクをイベント頻度の値
に基づいてソートし、前記温度の順位に対応するイベント頻度の順位を有するタ
スクを選択して割り当てる、

タスクスケジューリング装置。

10. 請求の範囲第9項において、

- 20 前記タスク割り当て部は、前記温度の順位が高いものからの順位である場合に
は、前記タスクを、低いイベント頻度から高いイベント頻度に向けてソートし、
前記温度の順位が低いものからの順位である場合には、前記タスクを、高いイベ
ント頻度から低いイベント頻度に向けてソートする、

タスクスケジューリング装置。

25

11. 請求の範囲第4項において、

- 前記記憶部に記憶された特性値は、タスクに含まれる命令の個数、単位時間当
たりに処理される前記命令の個数、タスクの実行時に行われるメモリへのアクセ
ス回数、単位時間当たりのメモリへのアクセス回数、前記命令と前記アクセス回
30 数との合計値、単位時間当たりの前記命令と前記アクセス回数との合計値、また

は前記タスクの処理に要する処理時間である，
タスクスケジューリング装置。

1 2．請求の範囲第5項から第11項のいずれか1項において，

5 前記命令は浮動小数点演算命令である，タスクスケジューリング装置。

1 3．請求の範囲第4から第12項のいずれか1項において，

前記タスクスケジューリング装置は，前記複数の処理装置の1つであり，自己
または他の処理装置に前記タスクスケジューリングを実行する，

10 タスクスケジューリング装置。

1 4．複数のタスクを分散して処理する複数の処理装置を有する分散処理システム
であって，

前記複数の処理装置のそれぞれの温度または消費電力を計測する計測装置と，
15 前記複数の処理装置とは別装置として設けられ，または，前記複数の処理装置
の少なくとも1つに設けられ，前記計測装置により計測された各処理装置の温度
または消費電力を比較し，前記比較の結果，前記計測装置により計測された温度
または消費電力が最も低い処理装置にタスクを割り当てるタスクスケジューリン
グ装置と，

20 を備えている分散処理システム。

1 5．複数のタスクを分散して処理する複数の処理装置を有する分散処理システム
であって，

前記複数の処理装置のそれぞれの温度または消費電力を計測する計測装置と，
25 各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関
連するタスクの特性値をタスクごとに記憶する記憶装置と，

タスクの割り当て対象となる処理装置についての，前記計測装置により計測さ
れた温度または消費電力と，前記記憶装置に記憶された前記特性値とに基づいて，
前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタス
クから選択し，該選択したタスクを前記タスクを割り当てる対象となる処理装置
30

に割り当てるタスク割り当て部と、
を備えている分散処理システム。

16. 複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または
消費電力を計測する計測装置とを有する分散処理システムにおける前記複数の処
理の少なくとも1つにより、または、前記複数の処理装置とは別個に設けられた
制御装置により実行されるタスクスケジューリング方法であって、
前記計測装置により計測された各処理装置の温度または消費電力を比較し、
前記比較の結果、前記計測装置により計測された温度または消費電力が最も低
い処理装置にタスクを割り当てる、
タスクスケジューリング方法。

17. 複数のタスクを分散して処理する複数の処理装置と各処理装置の温度または
消費電力を計測する計測装置とを有する分散処理システムにおける前記複数の処
理の少なくとも1つにより、または、前記複数の処理装置とは別個に設けられた
制御装置により実行されるタスクスケジューリング方法であって、
タスクの割り当て対象となる処理装置についての、前記計測装置により計測さ
れた温度または消費電力と、内部のメモリまたは外部の共有メモリに記憶され、
各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連
するタスクの特性値とに基づいて、前記タスクの割り当て対象となる処理装置に
割り当てるタスクを実行待ちのタスクから選択し、
該選択したタスクを前記タスクを割り当てる対象となる処理装置に割り当てる、
タスクスケジューリング方法。

18. 複数のタスクを分散して処理する複数の処理装置の少なくとも1つまたは前
記複数の処理装置とは別個に設けられた制御装置に設けられたコンピュータに、
各処理装置の温度または消費電力を計測する計測装置によって計測された前記
各処理装置の温度または消費電力を比較する手順と、
前記比較の結果、前記計測装置により計測された温度または消費電力が最も低
い処理装置にタスクを割り当てる手順と、

を実行させるためのプログラム。

19. 複数のタスクを分散して処理する複数の処理装置の少なくとも1つまたは前記複数の処理装置とは別個に設けられた制御装置に設けられたコンピュータに、
- 5 タスクの割り当て対象となる処理装置についての、計測装置により計測された温度または消費電力と、内部のメモリまたは外部の共有メモリに記憶され、各タスクの実行に伴う各処理装置の温度上昇または消費電力増加の度合いに関連するタスクの特性値とに基づいて、前記タスクの割り当て対象となる処理装置に割り当てるタスクを実行待ちのタスクから選択する手順と、
- 10 該選択したタスクを前記タスクを割り当てる対象となる処理装置に割り当てる手順と、
- を実行させるためのプログラム。

図 1

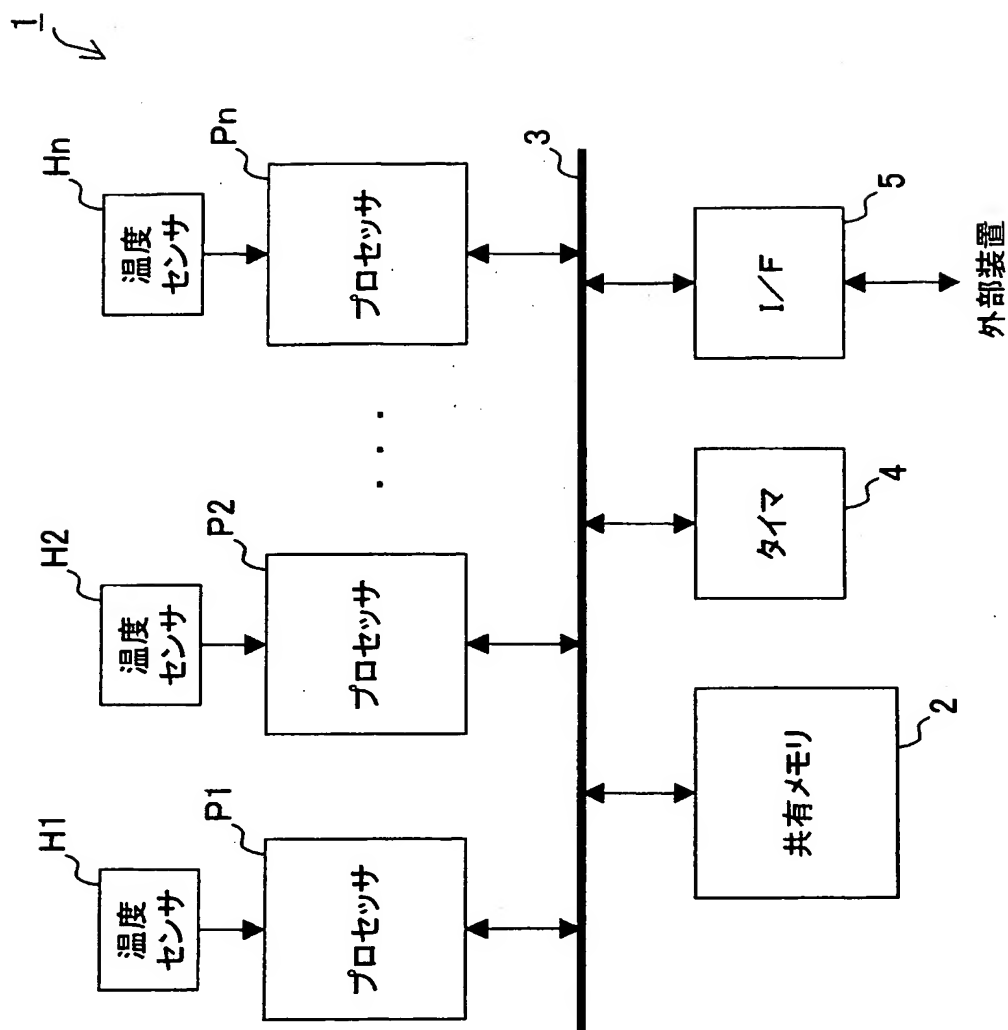


図 2

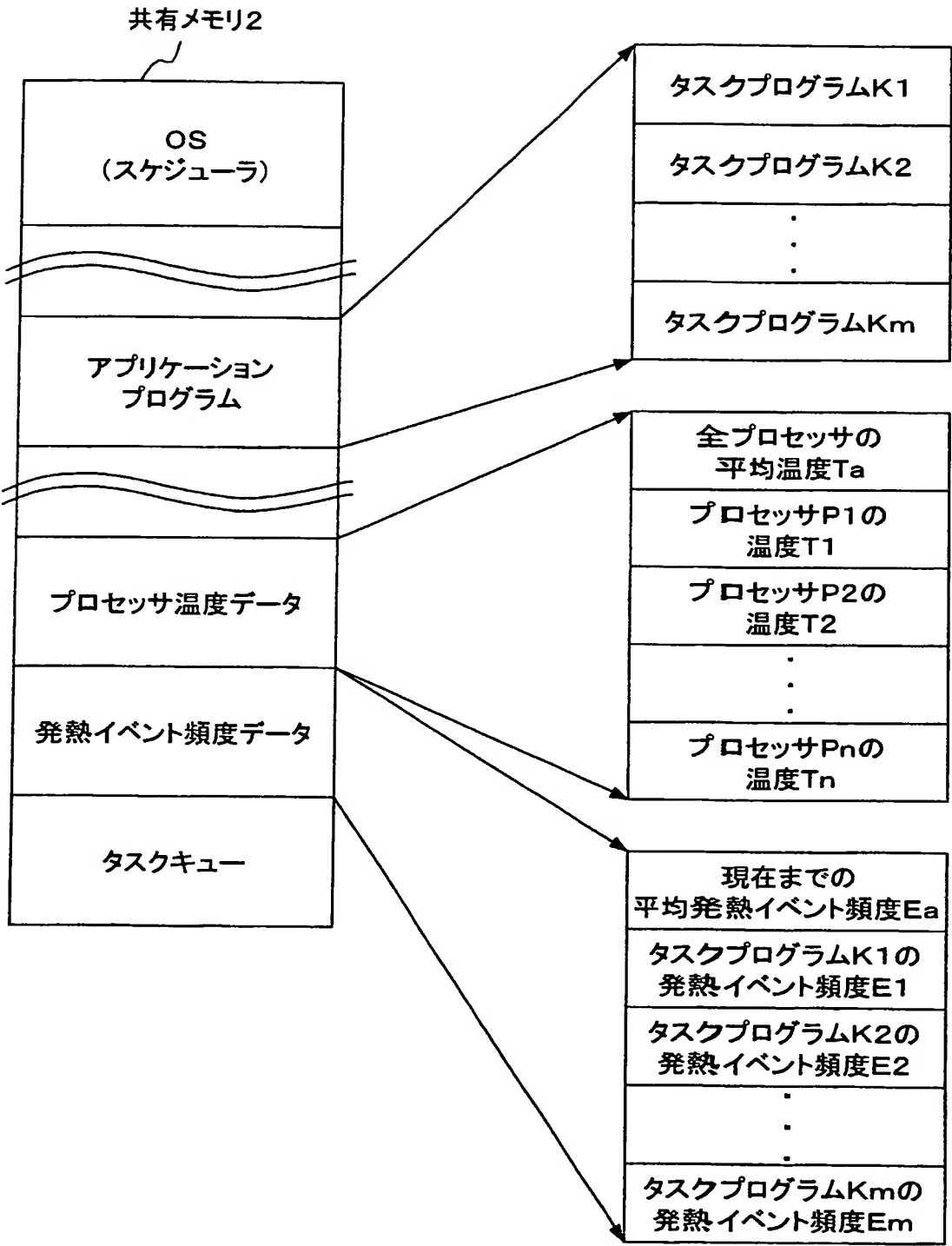


図 3

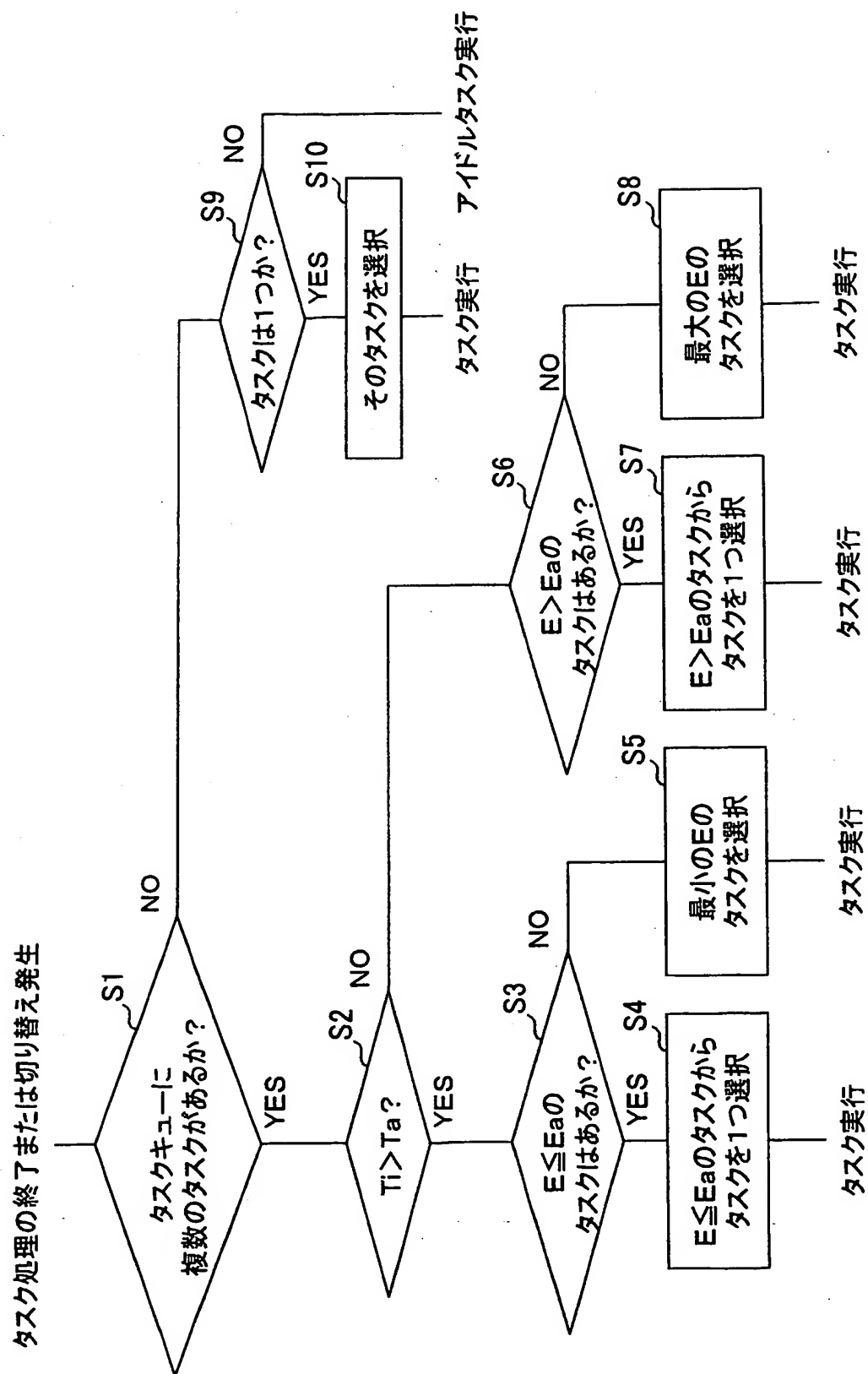


図 4

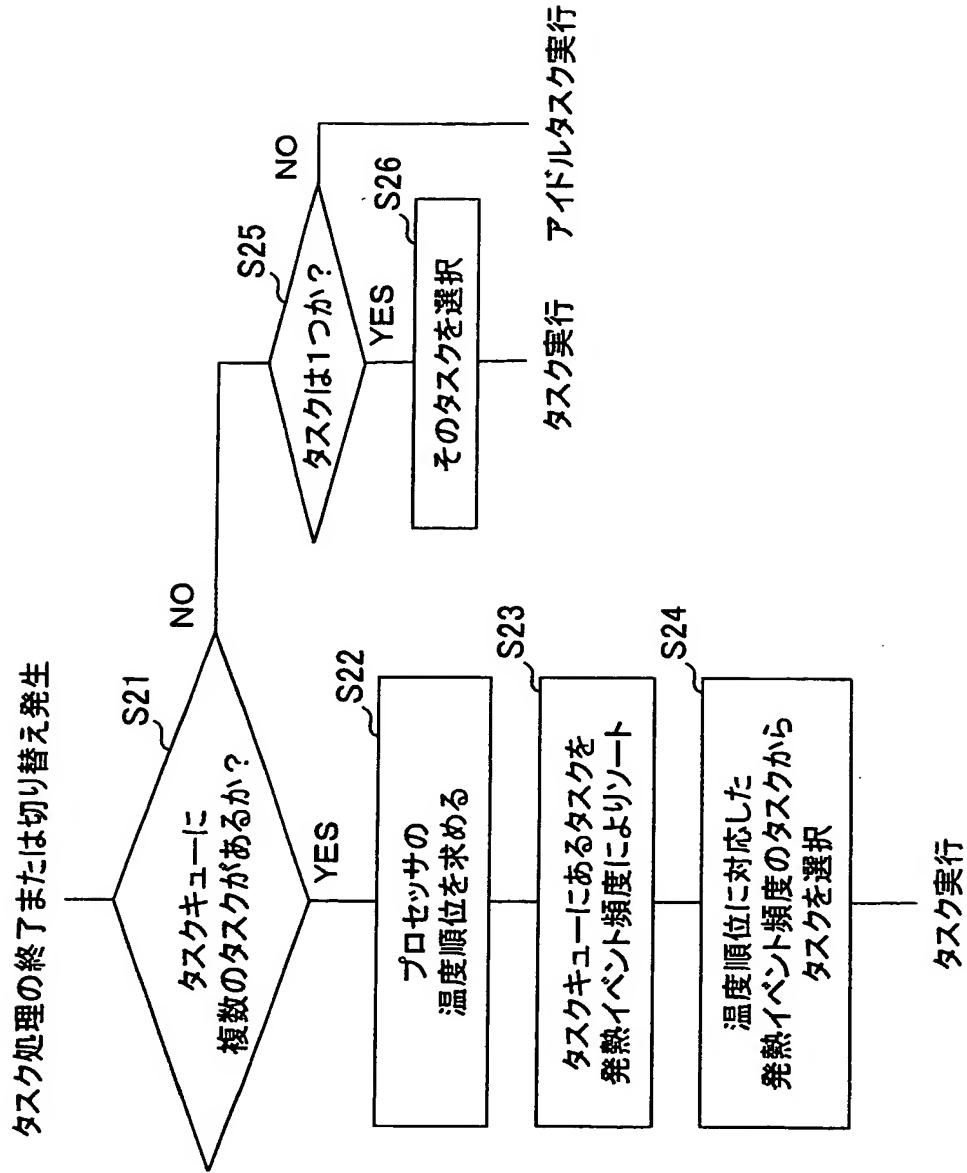
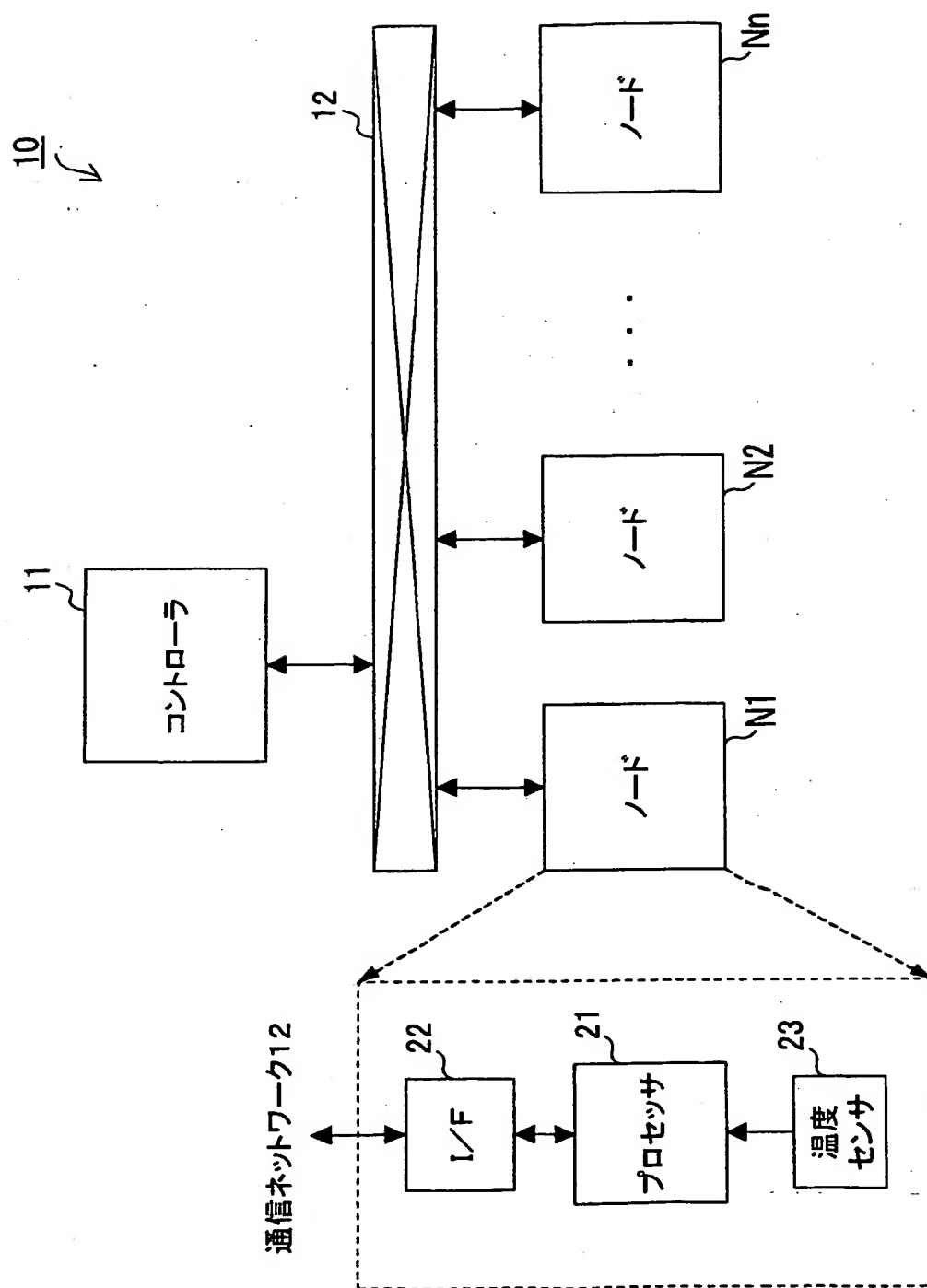


図 5



INTERNATIONAL SEARCH REPORT

International application No.

PCT/JP02/03324

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl⁷ G06F15/177, G06F9/46, G06F1/00

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

Int.Cl⁷ G06F15/16-15/177, G06F9/46, G06F1/00, G06F15/78

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Jitsuyo Shinan Koho	1922-1996	Toroku Jitsuyo Shinan Koho	1994-2002
Kokai Jitsuyo Shinan Koho	1971-2002	Jitsuyo Shinan Toroku Koho	1996-2002

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	JP 08-16531 A (Hitachi, Ltd.), 19 January, 1996 (19.01.96), Full text; all drawings (Family: none),	1-3, 14, 16, 18
A		4-13, 15, 17, 19
A	JP 11-296488 A (Hitachi, Ltd.), 29 October, 1999 (29.10.99), Full text; all drawings (Family: none)	1-19
A	JP 10-240704 A (Ricoh Co., Ltd.), 11 September, 1998 (11.09.98), Full text; all drawings (Family: none).	4-13, 15, 17, 19

☐ Further documents are listed in the continuation of Box C.☐ See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search
27 June, 2002 (27.06.02)Date of mailing of the international search report
09 July, 2002 (09.07.02)Name and mailing address of the ISA/
Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

Form PCT/ISA/210 (second sheet) (July 1998)

A. 発明の属する分野の分類 (国際特許分類 (IPC))

Int. Cl.⁷ G06F15/177, G06F9/46, G06F1/00

B. 調査を行った分野

調査を行った最小限資料 (国際特許分類 (IPC))

Int. Cl.⁷ G06F15/16-15/177, G06F9/46, G06F1/00, G06F15/78

最小限資料以外の資料で調査を行った分野に含まれるもの

日本国実用新案公報 1922-1996年
 日本国公開実用新案公報 1971-2002年
 日本国登録実用新案公報 1994-2002年
 日本国実用新案登録公報 1996-2002年

国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)

C. 関連すると認められる文献

引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X A	JP 08-16531 A (株式会社日立製作所) 1996.01.19, 全文, 全図 (ファミリーなし)	1-3, 14, 16, 18 4-13, 15, 17, 19
A	JP 11-296488 A (株式会社日立製作所) 1999.10.29, 全文, 全図 (ファミリーなし)	1-19
A	JP 10-240704 A (株式会社リコー) 1998.09.11, 全文, 全図 (ファミリーなし)	4-13, 15, 17, 19

☐ C欄の続きにも文献が列挙されている。☐ パテントファミリーに関する別紙を参照。

* 引用文献のカテゴリー

「A」 特に関連のある文献ではなく、一般的技術水準を示すもの
 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの
 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す)
 「O」 口頭による開示、使用、展示等に言及する文献
 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願

の日の後に公表された文献

「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの
 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの
 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの
 「&」 同一パテントファミリー文献

国際調査を完了した日

27.06.02

国際調査報告の発送日

09.07.02

国際調査機関の名称及びあて先

日本国特許庁 (ISA/JPO)

郵便番号100-8915

東京都千代田区霞が関三丁目4番3号

特許庁審査官 (権限のある職員)

久保 正典



5B

9642

電話番号 03-3581-1101 内線 3545

THIS PAGE BLANK (USPTO)